

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭61-96598

⑬ Int.Cl.⁴

G 11 C 17/00
H 03 K 21/00

識別記号

1 0 1

庁内整理番号

6549-5B
6749-5J

⑭ 公開 昭和61年(1986)5月15日

審査請求 未請求 発明の数 1 (全4頁)

⑮ 発明の名称 電氣的消去可能なP-R OMのカウンタデータ記憶方法

⑯ 特 願 昭59-218813

⑰ 出 願 昭59(1984)10月17日

⑱ 発 明 者 羽 生 裕 川崎市川崎区田辺新田1番1号 富士電機株式会社内
⑲ 出 願 人 富士電機株式会社 川崎市川崎区田辺新田1番1号
⑳ 代 理 人 弁理士 青山 葆 外2名

明 細 書

1. 発明の名称

電氣的消去可能なP-R OMのカウンタデータ
記憶方法

2. 特許請求の範囲

電氣的に書き換え可能な複数桁の記憶素子を有するP-R OMをカウンタデータ記憶手段として用いる場合において、カウンタデータのうち書き換え頻度の高い桁を分担していた記憶素子の書き換え度数が所定値になったとき、書き換え頻度の低い桁を分担するように、カウンタデータの桁と記憶素子の桁とを相対的にローテーションさせることを特徴とする電氣的消去可能なP-R OMのカウンタデータ記憶方法。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は電氣的消去可能なP-R OMを用いたカウンタデータ記憶方法に関する。

〔従来技術とその問題点〕

記憶データを電氣的に消去可能なP-R OMで

は、記憶素子のデータを書き換える度に記憶素子のスレッショルド電圧幅が変化する為、最終的には記憶素子は永久に消去されたままになり、データ書き換えの繰返し回数に限度がある。このP-R OMを電子カウンタのカウンタデータ記憶手段として使用した場合、使用頻度の最も高い1の位(10⁰桁)を制御する素子は一希先に消去限度回数を越え寿命が尽きる。これを防ぐ為に、1の位に関しては、予備の記憶素子を用意しておくという方法も考えられるが、この方法では、メモリ領域がその分だけ減少し、P-R OMの持つ機能を有効に生かす事が出来ないという問題があった。

〔発明の目的〕

本発明は、各記憶素子を使用頻度の異なる各桁間に順次ローテーションさせることにより、各記憶素子を寿命近くまで有効に利用できるP-R OMのカウンタデータ記憶方法を提供することを目的とする。

〔発明の構成〕

本発明の電氣的消去可能なP-R OMのカウ

トデータ記憶方法は電氣的に書き換え可能な、複数桁の記憶素子を有するP-ROMをカウントデータ記憶手段として用いる場合において、カウントデータのうち書き換え頻度の高い桁を分担していた記憶素子の書き換え度数が所定値になったとき、書き換え頻度の低い桁を分担するように、カウントデータの桁と記憶素子の桁とを相対的にローテーションさせることを特徴とする。

[実施例]

以下に、この発明の1実施例を図面とともに説明する。

第1図において、1はカウントデータ記憶手段としての電氣的に書き換え可能なP-ROMであり、 $b_0 \sim b_n$ はカウントデータの各桁の記憶素子を示している。

2はCPUでP-ROM1へのカウントデータの書き込みを制御するとともに、後述するようなP-ROMの記憶桁のローテーションを制御する。

3はローテーションの回数を計数するカウンタで、このカウンタも電氣的に書き換え可能なP-ROM

Mを用いて構成することができる。4はメモリ(RAM)である。

P-ROM1の各素子への書き込み・消去が $m \cdot 10^n + \alpha$ 回 (m, n は自然数, $\alpha < 10^n$)

で不能となる場合(例えば12万回で不能となる場合、 $m=1, n=5, \alpha=20000$ である。

)について、以下第2図と第3図を参照して本発明をさらに詳細に説明する。

ステップS1では初期化としてP-ROM1の各桁 $b_0 \sim b_n$ をすべて0にリセットする。

ステップS2でP-ROM1はCPU2から印加される信号によってカウント動作を始め、カウントアップする。

なお、この実施例においては、P-ROM1の使用開始直後は図上右端の素子 b_n が、カウントデータの1の位(10^0 桁)、以後左へ順に b_{n-1}, b_{n-2}, \dots が $10^1, 10^2, \dots$ の位($10^1, 10^2, \dots$ 桁)をそれぞれ記憶するものとする。

ステップS3では 10^n 桁、この実施例では b_5 のデータが m の整数倍になったかどうか(変化し

た時点)をCPU2が判断する。

例えば、P-ROM1の各素子への書き込み・消去が12万回で限度とされる場合には12万回 = $1 \cdot 10^5 + 20000$ 回なので $m=1$ に設定しておく、こうしておく事で、 10^5 桁目すなわち1桁目が $m \times 10^n$ 回動作したかどうかを検知する。

ここで、 10^n 桁のデータが m の整数倍(カウントデータ = $m \times 10^n$)であれば、即ち素子 b_n が「1」になるとステップS4に進み、P-ROM1の各桁を左へ1桁分だけローテーションする。

第3図にローテーションの例を示す。

即ち、P-ROM1の素子 b_0 が0から1に変わったことをCPU2が検出すると、P-ROM1の素子 b_0 が 10^1 の桁、 b_1 は 10^2 の桁、素子 b_2 は 10^3 の桁というように各素子が記憶するデータの桁のローテーションが行われる。

即ち 10^n 桁を記憶する素子 b_n の書き換えが消去不能となる回数近くなった事(第2図の例では12万回の書き換えで寿命となる素子を使用する場合には10万回で他の素子に代替される。)

を検知し、使用頻度の少ない素子 b_0 を使用頻度の多い 10^0 桁目を記憶させる事にある。

このとき、限界近くまで使用された 10^0 桁目を記憶していた素子 b_0 は最も使用頻度の少ない最上位(10^n 桁)を記憶する様にする。

ステップS5では、上述のローテーション1回行なう毎にカウンタ3が1ずつカウントし、ステップS6でP-ROM1をクリアし、ステップS5でのカウンタ3のカウント値に $m \times 10^n$ (この実施例では 1×10^5) を乗じその結果をステップS7にて、いったんメモリ4へ格納する。

そしてステップS8ではその演算結果(即ち、現在のカウントデータ)をP-ROM1に(記憶)表示する。その後はステップS2に戻り、再びカウントアップを始める。このときは素子 b_n が 10^0 桁を分担する。

尚、ステップS8において、カウントデータ(P-ROM1の記憶内容)は、1回目では $m \times 10^n$ 、2回目では $2m \times 10^n$ 、3回目では $3m \times 10^n$ となる。

この発明では各桁を記憶する素子を、書き込み・消去が不能となる回数のある程度以下で使用する事が前提となっている。

このある程度以下を表すものが、最初に述べた $n \cdot 10^n + \alpha$ の α である。例えば6桁のカウンタで1桁目が 10^5 回動作したらローテーションするようにしたとする。

この時、ローテーションを5回すなわち、各桁を記憶する素子を均等に使用した場合、その素子の書き込み・消去回数は $10^5 + 10^4 + 10^3 + 10^2 + 10^1 + 10^0 = 1111,111$ 回となる。すなわち、この方法(1桁目が 10^5 回動作したらローテーションする方法)は、P-R-OMの書き込み消去回数が111111回以上の物($\alpha > 111111 = 111,111 - 100000$)でなければならない。

この様に、 α 値の設定には制約がある。

P-R-OMへの書き込み・消去限度をX回とするとXは、次式(1)で表わす事ができる。

$$X = n \cdot 10^n + \alpha \quad (n, \alpha \text{ は自然数}, \alpha < 10^n) \quad \dots (1)$$

ここで、 10^n 桁のデータが n になったら各桁をローテーションするソフトウェアに於いて、 n 回ローテーションした結果各素子への書き込み・消去は、次式(2)で表わされる回数分行われた事になる。

$$\text{書き込み消去回数 } N = n \times (10^n + 10^{n-1} + 10^{n-2} + \dots + 10^0) \text{ 回} \quad \dots (2)$$

各桁をローテーションする前に素子がグロウンしてしまわない為には、P-R-OMへの書き込み・消去限度回数Xは、実際の書き込み・消去回数Nより大きくなければならない。

$$X - N > 0 \quad \dots (3)$$

(1), (2), (3)式より

$$\alpha > n(10^{n-1} + 10^{n-2} + \dots + 10^0) \quad \dots (4)$$

α の値は上式(4)を満たす必要があり、上式を満足できないときは n の値を変えればよい。

上記の方法によれば各素子の使用頻度の均一化を計ることができる。

「発明の効果」

本発明によれば、P-R-OMの素子への書き込み限度回数を設定する事でカウントデータの各桁を記憶する各素子について、ある素子が書き込み限度に達すると、ローテーションをして、過去において書き換え頻度の低かった素子には、書き換え頻度の高い桁を分担させ、逆に過去において書き換え頻度の高かった素子には、書き換え頻度の低い桁を分担させることにより、上記のローテーションを桁数分だけ行なってカウントデータを記憶することができるので、P-R-OMの各素子を均一に動作させてより多くの数をカウントする事ができ、非常に効率的である。

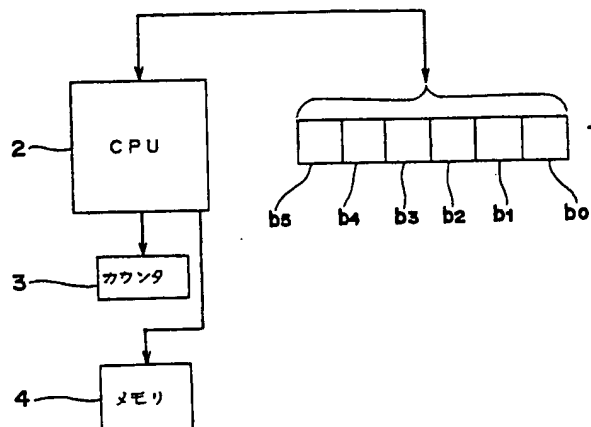
1. 図面の簡単な説明

第1図は本発明の1実施例を示すブロック図、第2図は第1図の実施例の動作を示すフローチャート図、第3図は第2図における動作を示す図である。

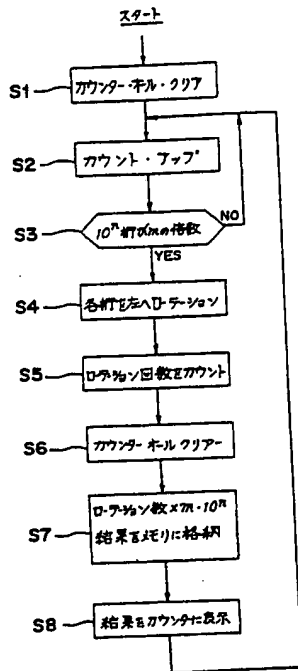
1…P-R-OM、2…CPU、

3…ローテーションカウンタ、4…メモリ。

第1図



第 2 図



第 3 図

